

First Hit
☐ [Generate Collection](#) [Print](#)

L30: Entry 59 of 212

File: PGPB

Feb 20, 2003

DOCUMENT-IDENTIFIER: US 20030037048 A1

TITLE: METHOD AND APPARATUS FOR PARALLEL EXECUTION OF SQL-FROM WITHIN USER DEFINED FUNCTIONS

Detail Description Paragraph:

[0053] The query composer 508 allows a user to compose a SQL query using a simple text editor. The RPC 114 is the interface to the query coordinator 104. It ships SQL queries to the query coordinator 104 for execution and retrieves result tuples into the cache 510. The cache 510 comprises a master data cache 510A, a metadata cache 510B and an object cache 510C. The object cache 510C caches the result of a query in formats understood by the map view 502. The metadata cache 510B stores the catalog information of the currently open database. The master data cache 510A stores retrieved master data as described further below. In one embodiment, the object cache 510C also caches the objects downloaded from the data servers 130. FIG. 5 also shows the second communication path 140 from the data server 130 to the user front end 108 via the direct data transfer module 142.

Detail Description Paragraph:

[0079] Parallel execution of UDFs relies on the fact that all UDFs are executed on the various data servers 130 in the parallel system 100. In other words, the UDFs are associated with and executed on particular data servers 130. As described above, it is desirable to embed and utilize SQL statements from within UDFs. Further, it is desirable that the parallelization and execution of the SQL remains transparent to the UDF such that the user defining the UDF does not know, need to know, or maintain knowledge of the underlying parallelization of the DBMS.

Detail Description Paragraph:

[0105] FIG. 9 is a flow chart illustrating the retrieval of query results in accordance with one or more embodiments of the invention. At step 900, the various dispatcher instances 204A-204E output their results to a single dispatcher instance 204. At step 902, the single dispatcher instance 204 (at a particular data server 130) merges the results from the parallel execution of the various data servers 130 to produce a stream of tuples. At step 904 the merged stream of tuples are output to the specified endpoint. At step 906, the UDF obtains an input stream from the endpoint. At step 908, each tuple may be obtained from the input stream by the UDF. Execution of the UDF continues at step 910 (e.g., the UDF can perform operations on the tuple or continue executing other SQL or non-SQL commands).

Detail Description Paragraph:

[0110] Generally, each node may operate under the control of an operating system stored in memory. In one or more embodiments, the client program 102 (and the other nodes) interfaces with the user to accept inputs and commands and to present results through a graphical user interface (GUI) (e.g., user front-end 108). The instructions performing the GUI functions can be resident or distributed in the operating system, a computer program, or implemented with special purpose memory and processors. Embodiments of the invention may also implement a compiler that allows an application program written in a programming language such as COBOL, C++, FORTRAN, or other language to be translated into processor readable code. After completion, the application may access and manipulate data stored in memory using the relationships and logic that was generated using the compiler. Embodiments of

the invention also optionally comprise an external communication device such as a modem, satellite link, Ethernet card, or other device for communicating with other computers.